# A SOFTWARE ARCI IITECTURE
# FOR AUTOMATING OPIINA"1'1ONS PROCESSES

KEVIN MII.LER
Contact: SUSAN MURPHY
Operation Engineering 1.ab
Jet Propulsion Laboratory, MS 301-345
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109-8099
(81 8) 354-1948
I'AX: (818) 393-41OO
susan@natasha.jpl.nasa.gov

The Operations Engineering Lab (OEL) at JPL has developed a software architecture based on an integrated toolkit approach for simplifying and automating mission operations tasks. The toolkit approach is based on building adaptable, reusable graphical tools that arc integrated through a combination of libraries, scripts, and systcm-level user interface shells. 'I'he graphical interface shells arc designed to integrate and visually guide a user through the complex steps in an operations process. They provide a user with an integrated syslcm-level picture of an overall process, defining the required inputs and possible outputs through interactive on-screen graphics.

The OEL has developed the software for building these process-oriented graphical user interface (GUI) shells. The 0111. Shell development system (OELSHELL) is an extension of JPL's Widget Creation Library (WCI .). The OELSHELL system can be used to easily build user interfaces for running complex processes, applications with extensive command-line interfaces, and tool-integration tasks. The interface shells display a logical process flow using arrows and box graphics. They also allow a user to select which output products arc desired and which input sources arc nccdcd, eliminating the need to know which program and its associated command-line parameters must be executed in each case. The shells have also proved valuable for usc as operations training tools because of the OEI .S1 IEI.L hypertext help environment.

The OEL toolkit approach is guided by several principles, including the usc of ASCII text file interfaces with a multimission format, Per] scripts for mission-specific adaptation code, and programs that include a simple command-line interface for batch mode processing. Projects can adapt the interface. shells by simple changes to the resource configuration file. 'J'his approach has allowed the development of sophisticated, automated software systems that arc easy, cheap, and fast to build.

This paper will discuss our toolkit approach and the 01;1.S111?1,1, interface builder in the context of a real operations process example. The paper will discuss the design and implementation of a Ulysses toolkit for generating the mission sequence of events. The Sequence of Events Generation (SEG) system provides an adaptable multimission toolkit for producing a iimc-ordered listing and time.line display of spacecraft commands, state changes, and required ground activities. The multimission SEG software is easily adapted and 01;1.S111~1 .L templates arc built to meet different mission requirements. The SEG system was adapted in a unique way for the Ulysses mission since the spacecraft does all commanding in real time. The Ulysses SEG toolkit allows a user to interactively build commands on a timeline display in spacecraft event time and then the system automatically derives required ground events, builds a mission sequence of e.vents listing, and outputs a space flight operations schedule.